Chapter 3

Methodology

3.1 Evaluation

We use CROW, the tool described at section 2.1, to answer RQ1. This section describes the corpora of original programs that we pass to CROW for the sake of variants generation. Besides, we describe our metrics and finalize the section by discussing the results.

3.1.1 Corpora

We answer RQ1 with two corpora of programs appropriate for our experiments. The first corpus, **CROW prime**, is part of the CROW contribution []. The second corpus, **MEWE prime**, is part of the MEWE contribution []. In Table 3.1 we summarize the selection criteria, and we mention each corpus properties. With both corpora we evaluate CROW with a total of $303 + 2527$ functions.

Corpus name	No. programs	No. functions	LOC range	Location
Rosetta	303	303	7 - 150	https://google.com
Libsodium	5	687	TODO	https://google.com
QrCode	2	TODO	TODO	https://google.com
Total	310	TODO	+ TODO	TODO

Table 3.1: TODO

3.1.2 Metric

To assess our approach's ability to generate WebAssembly binaries that are statically different, we compute the number of unique variants generated by CROW

for each original function. We compare the WebAssembly program and its variant using the `md5` hash of each function byte stream as a metric for uniqueness.
TODO Move text out of the table

3.1.3 Setup

CROW's workflow synthesizes program variants with an enumerative strategy. All possible programs that can be generated for a given language (LLVM in the case) are constructed and verified for equivalence. There are two parameters to control the size of the search space and hence the time required to traverse it. On the one hand, one can limit the size of the variants. On the other hand, one can limit the set of instructions used for the synthesis. On the other hand, in our experiments, we use between 1 instruction (only additions) and 60 instructions (all supported instructions in the synthesizer).

These two configuration parameters allow the user to find a trade-off between the number of variants that are synthesized and the time taken to produce them. In Table 3.2 we listed the configuration for both corpora. For the current evaluation, given the size of the corpus, we set the exploration time to 1 hour maximum per function for **CROW PRIME**. In the case of **MEWE prime**, we set the timeout to 5 minutes per function in the exploration stage. We set all 60 supported instructions in CROW for both **CROW prime** and **MEWE primer** corpora.

CORPUS	Exploration timeout	Max. instructions
CROW prime	1h	60
MEWE prime	5m	60

Table 3.2: CROW tweaking for variants generation. The table is composed by the name of the corpus, the timeout parameter and the count of allowed instructions during the synthesis process.